경영학연구논문집 제5권 제1호 pp. 1~16, 울산대학교.
Journal of Mangement Vol 5, No. 1, pp 1~16, 1998
University of Ulsan

k-링크 장애시의 통신망 생존도 산출방법에 관한 연구

김현준
경영학부

<요 약>

최근 광통신기술을 바탕으로 통신망의 교환 및 전송설비들이 대용량화함에 따라, 통신망의 구성요소 장애에의 대처능력 분석이 주요연구대상이 되고 있다. 본 연구에서는 전송설비인 링크의 장애에 대처하는 능력으로서 통신망의 링크생존도를 분석한다. 먼저 k개의 전송링크에 동시적인 장애가 발생한 경우에도 여전히 처리될 수 있는 서비스 양의 비율을 k-링크 생존도로 정의하고, 그 산출의 수리적 특성을 분석한다. 이를 토대로, 링크의 이중장애 및 다중장애의 경우에 일반적으로 적용될 수 있는 생존도 산출 방법론을 제시하며, 제시된 방법론의 계산상의 복잡도를 분석한다. 또한 다양한 규모의 그래프를 대상으로 이루어진 실험결과를 통하여 제안된 방법론이 현실에서도 효과적으로 적용될 수 있음을 보이고 있다.

A Computation Procedure for the k-Link Survivability

Hyun-joon Kim
Professor of Management Information Systems

<Abstract>

As the fiber optic technology enlarges the capacity of switching and transmission facilities, a special emphasis is placed on the analysis of networks' capability coping with component failures. This study deals with the k-link survivability of networks, which is defined as the relative ratio of service requirements still intact under simultaneous failures of k links. With the analysis of mathematical characteristics of the link survivability, a computational procedure for the k-link survivability is presented. Additionally, the computational complexity is investigated and computational results are summarized, which show that the proposed procedure could be applied to the analysis of large real-world networks.

I. 서 론

광섬유를 이용한 정보전송이 본격화되면서 정보통신망은 기능, 용량 및 응용분야의 측면에서 획기적인 변화를 보여주고 있다. 물론 이는 기술면에서 전송장비로서의 광섬유, 교환설비로서의 ATM교환기, 정보단말인 컴퓨터의 정보처리능력이 획기적으로 발전함에 따른 것이며, 사회의 정보화에 따른 다양한 정보서비스에 대한 수요의 창출에 힘입은 바가 크다 하겠다 또, 다양한 정보의 디지틀화에 따른 멀티미디어 처리기술의 발달에 따라 인터넷 서비스의 폭발적인 증가도 통신망의 고도화 및 광대역화를 촉진하는 계기가 되고 있다. 어쨌든 대용량의 빠른 교환/전송설비의 도입은 통신망의 구조적인 변화를 수반하게 되었으며, 이러한 변화는 망구조의 단순화로 나타나게 되었다 즉, 동축케이블에 의존하던 과거에는 노드 설비간에 다수의 전송회선이 설치되어야만 했고, 그리 크지 않은 전송설비의 용량으로 인하여 매우 복잡한 그물모양의 통신망이 주를 이루었으나, 최근 또는 향후의 통신망은 매우 큰 용량을 갖는 전송설비의 특성에 따라 노드 설비들을 매우 적은 수의 회선으로 연결시키는 것만으로도 충분한 정보처리기능을 제공하게 된 것이다(Wu, 1992).

한편, 이러한 변화와 더불어 사회의 정보의존 및 통신의존은 심화되었고, 결과적으로 통신망의 성능평가 지표의 비중도 변하게 되었다 즉, 과거의 복잡한 그물형태의 망에서는 망설비의 기능적 품질을 평가하기 위한 신뢰도나 서비스 요구를 충족시킬 수 있는 서비스의 가용도의 측면이 부각되어 관리되던 것이, 단순화되는 망에서는 망설비의 일부분에 장애가 발생했을 때에도 여전히 주어진 서비스 요구를 원활히 처리할 수 있느냐 하는 생존도 개념위주의 평가로 바뀌어 가고 있는 것이다 이는 사회의 정보화에 따라 통신망에 부과된 정보의 양이 많아짐과 동시에 정보의 유실에 따른 파급손실이 급증하는 현상(즉, 정보의 상대적 가치가 커진 현상)을 반영하는 것이라 볼 수 있다. 현실에서의 이러한 흐름에 발맞추어 80년대 말 이후 통신망의 생존도 개념은 학계에서 많은 주목을 받았으며, 망의 생존도 분석이나 생존도를 강화시킬 수 있는 통신망 설계방법들이 활발히 연구되기 시작하였다(Wu, Kolar & Cardwell, 1988, Cardwell, Monma & Wu, 1989; Cardwell & Brush, 1990).

기존 연구들이 근거하고 있는 통신망의 생존도는 크게 두 가지로 대별된다. 그 하나는 망에 부과되는 서비스 요구량과는 무관하게 망의 노드간 연결성을 강화시키는 것이 망요소 장애시에도 서비스의 원활한 제공을 보장한다는 시각하에서, 망의 생존도를 노드간의 연결성으로 규정하여 접근하는 시각이다. 다른 하나는 생존도 개념이 궁극적으로 서비스의 원활한 제공에 그 목적이 있는 만큼, 망요소 장애시에 단절되는 서비스 요구를 최소화하는 것이 생존도 제고의 적정한 수단이라는 시각으로, 주어진 서비스 요구를 정량적으로 반영하려는 시도이다. 전자의 연결성 위주의 접근은 서비스 요구와는 무관하게 통신망의 구조 및 설비 용량만을 중심으로 접근하므로 이론적 정형화가 용이하며 기존에 활발히 연구되었던 그래프이론들을 쉽게 접목시킬 수 있는 이점이 있었으므로, 우선적으로 연구되어지고 있다(Gavish et. al, 1989; Monma & Shallcross, 1989; Monma, Munson & Pullyblank, 1990; Grötchel, Monma & Stoer, 1992). 다만 이러한 접근은 주어지는 서비스 요구량에 근거하지 못하므로, 노드간 연결성에 근거한 통신망의 구축은 망서비스 제공에 따르는 적절한 경제성 확보가 어렵다는 비판을 받을 수 있다. 즉, 노드간의 연결성을 어느 정도로 규정할 것인지를 객관적인 지표에 근거하여 도출하기에 어려움이 있기 때문에, 결과적인 망의 생존도와 망구축비용의 효율적인 균형을 도모하기가 어렵고, 이는 결국 망구축비용의 과도화로 이어질 수 있게 되는 것이다

통신망의 생존도를 정량적으로 분석하려는 접근은 1980년대 후반부터 본격화되었다(Wu, Kolar & Cardwell, 1988; Brush & Marlow, 1990; Kim, 1995) 망구조에 따라 장애발생시의 서비스 처리능력이 달라지는 현상의 분석과 이에 대비하는 기술적인 방안의 연구들(Cardwell, Monma & Wu, 1989; Wu, Kolar & Cardwell, 1988)을 토대로 객관적인 생존도 지표의 개념이 구체화되었고, 이후 제한된 범위에서나마 생존도 산출방법이 제시되었다(Kim, 1995). 최근에는 이러한 정량적인 생존도 개념에 근거한 통신망의 설계모형 및 해법이 제시되고 있다(Wu, 1992; Myung, Kim & Tcha, 1997). 이들 연구에서 사용되는 통신망의 생존도는 "망의 일부 구성요소에 장애가 발생하는 경우에도 여전히 처리될 수 있는 서비스 요구량의 상대적인 비율"로 정의되며, 현재까지는 장애설비의 범위를 노드간의 링크로 국한시키고 있는 상태이다 이와 같은 통신망의 정량적 생존도는 망의 구조만이 아닌 주어지는 서비스 처리 요구량을 동시에 감안하기 때문에 수리적인 모형 또는 분석틀로의 정형화가 어렵다는 약점을 가지고 있으나, 보다 현실적인 개념에 근거하므로 그 응용성은 높다고 하겠다. 특히 최근의 연구결과는 정량적인 생존도에 근거한 망설계 모형이 노드간의 연결성에 근거한 정성적 망설계모형을 일반화시키고 있음을 보여주고 있다(Myung, Kim & Tcha, 1997) 따라서 향후의 정량적 생존도 관련 연구는 보다 활발해 질 것이며, 이를 위한 관련연구들이 심도 있게 연구될 필요가 있다 하겠다.

본 연구에서는 망구성요소 중의 링크설비 장애에 따른 정량적 망생존도를 분석한다. 즉, 몇 개의 링크설비에 장애가 발생하여 그 전송능력이 상실되는 경우에 어느 정도의 서비스 요구가 여전히 충족될 수 있는가를 산출하는 체계적 방법을 제시하고자 한다. 정량적 생존도로서 k-링크 생존도는 k개의 링크에 장애가 발생했을 때에도 여전히 처리되어지는 서비스 요구량의 상대적 비율로서 정의된다. 이러한 정의는 이미 기존의 연구에서 정형화된 것이며, 단일링크 생존도 및 이중링크 생존도는 이미 그 산출방법까지 체계화되어 있다(Kim, 1995). 그러나 일반화된 k-링크장애시의 생존도 산출방법은 구체적으로 제시된 바 없으며, 개념적 절차만이 도입되었으나 시험적용은 이중링크의 장애까지만 이루어졌다(Myung, Kim & Tcha, 1997) Kim(1995)이 제시한 싸이클 분해법도 단일링크 및 이중링크 생존도에만 적용 가능하다. 이에 본 연구에서는 일반화된 k-링크장애시의 생존도 산출절차를 체계화하고, 그 계산상의 복잡도 분석, 이중링크 생존도를 대상으로 한 기존의 싸이클 분해법과의 비교 등을 행하고자 한다

2장에서는 정량적 생존도의 개념 및 특성분석, 싸이클 분해법 등을 정리하고, 본 연구의 배경을 구체화한다. 3장에서는 일반화된 k-링크 생존도 산출의 이론적 토대와 알고리즘적 구현을 제시하며 계산상의 복잡도를 분석한다. 4장에서는 현실에서 운용되는 규모 정도의 통신망을 대상으로 제시된 방법론의 시험적용과 그 결과를 정리한다. 기존 싸이클 분해법과의 비교를 통하여 제시된 방법론의 현실적 적용가능성을 검증하며, 아울러 대상문제의 체계적 산출방법도 소개한다.

II. k-링크 생존도의 특성분석

본 장에서는 k-링크 생존도의 수리적인 정의와 특징을 분석한다. 또한 기존에 제시되어 있는 이중링크 생존도 산출을 위한 싸이클 분해법의 개요를 정리하고, 그 확장상의 한계를 지적하며, k-링크 생존도 산출을 위해서 필요한 기존 연구를 정리한다

2.1 k-링크 생존도의 수리적 정의

생존도 분석의 대상이 되는 통신망은 노드와 링크로 구성되는 그래프와 노드간에 부과되는 서비스 요구량으로 특징 지워진다 일반적인 그래프 G는 노드 집합을 $N=\{1,\cdots,n\}$, 링크집합을 $E=\{(i,j)|i\in N, j\in N\}$라 할 때, $G=(N,E)$로 표현된다

한편, 망에 부과되는 서비스 요구는 $T=\{t_{ij} \mid i\in N,\ j\in N\}$로 표현할 수 있으므로, 일반적인 통신망은 (G,T)로 표현될 수 있다 또 정량적 생존도 분석에 있어서 서비스 요구는 종단 노드간의 단위시간당 정보전송량을 의미하며, 링크설비의 장애는 두 노드간의 전송전송 불가능을 의미하기 때문에 링크의 방향성은 고려하지 않는다

다음으로 사용되는 부호를 정의한나. 임의의 그래프에서 링크의 축약(contraction)은 양쪽의 노드를 하나로 합치고 해당 링크를 삭제하는 것을 의미한다 이 때, 양쪽 종단 노드 중의 하나와 관련되는 서비스 요구는 통합된 노드로의 서비스요구로 전환되며, 양쪽 종단 노드간의 서비스 요구는 삭제된다 한편, 일부 링크를 삭제했을 때, 결과적으로 그래프내의 노드들이 분할되는 경우, 이들 링크의 집합을 컷(cut)이라 한다 그래프내의 두 노드를 연결시켜주는 연속되는 링크의 집합을 경로(path)라 하며, 첫 노드와 끝 노드가 동일한 경로를 싸이클(cycle)이라 한다. 그래프 G의 노드 또는 링크의 일부분만으로 구성되는 그래프를 그래프 G의 하위그래프(subgraph)라 하며, G에 속하는 두 노드 x, y간에 존재하며 중복되는 링크를 포함하지 않는 경로의 수를 x-y링크연결도라 하고, $\lambda(x,y,G)$로 나타내기로 한다 또 $\lambda(G)$는 그래프 G의 링크연결도이며, G내의 임의의 두 노드 사이에 존재하는 중복되지 않는 경로의 최소 수를 나타낸다 마지막으로 $\delta(G)$는 그래프 G에 포함되는 노드의 degree중에서 최소값을 나타낸다고 하자

한편, 그래프 G와 노드간 서비스 요구 T를 가지는 통신망에서 k-링크 생존도는 k개의 링크에 장애가 발생한 경우에도 여전히 처리되어질 수 있는 서비스의 상대적인 비율을 의미하게 되므로, 통신망 (G,T)의 k-링크 생존도 $S_k(G,T)$는 다음과 같이 표현된다.

$$S_k(G,T)=\frac{\sum_{i\in N}\sum_{j\in N}t_{ij}-W_k(G,T)}{\sum_{i\in N}\sum_{j\in N}t_{ij}}.$$

여기서 $W_k(G,T)$는 k개의 링크에 동시적인 장애가 발생한 경우에 차단되는 서비스 요구량의 최대량을 의미한다. 이렇게 정의하면, $S_k(G,T)$를 구하는 문제는 $W_k(G,T)$로 귀착되게 된다.

$W_k(G,T)$는 기본적으로 모든 k-링크 컷에 대하여 그 단절에 따른 서비스의 손실을

계산해 봄으로써 산출될 수 있다. 그러나 그래프내의 k-링크집합들 중에서 컷만을 체계적으로 나열하는 것은 쉽지 않기 때문에, 결국 모든 k-링크집합 전체를 대상으로 그 단절에 따른 서비스 손실량을 계산하여야만 한다. 즉, 임의의 k-링크집합에 대해서 그 단절이 그래프내의 노드를 분할하는지 파악하고, 노드를 분할하는 경우에 그에 따른 서비스의 손실량을 모두 조사하여야만 하는 것이다. 이와 같은 열거적인 개념에 근거한 생존도 계산절차는 적지 않은 계산상의 부담을 초래한다. 즉, 그래프 내의 링크 수를 m이라 하면, k-링크집합의 개수는 ${}_mC_k$이며, 각각의 단절에 따라 최대 ${}_nC_2$의 서비스 요구량에 대한 계산을 하게 되므로 전체적으로 $O(m^k n^2)$의 수학적인 복잡도를 가지게 된다. 결국 k-링크 생존도 산출의 문제는 어떻게 효과적으로 k-링크 컷들을 나열하고 그에 따른 서비스 단절량을 산출하느냐의 문제로 귀결되며, 그래프의 구조와 컷과의 관계를 어떻게 이용하느냐에 따라서 절차상의 효율성이 달라지게 된다.

2.2 이중링크 장애시의 생존도 산출

현재 링크에의 동시적인 장애발생 가능성을 2개까지로 제한하여 싸이클 분해법(cycle factoring method)을 이용하는 효과적인 이중링크 생존도 산출절차가 제시되어 있다(Kim, 1995). 이는 '컷을 이루는 두 개의 링크는 특정의 싸이클에 모두 포함되거나 아니면 모두 포함되지 않는다'는 그래프 이론에 근거하고 있다(Swamy & Thulasiraman, 1981). 즉, 그래프내의 노드들을 분할시키는 두 개의 링크가 있을 때, 그 중의 하나만을 포함하는 싸이클은 존재할 수 없다는 것이며, 따라서 그래프 G내의 임의의 싸이클 C을 찾고, 이로부터 해당싸이클의 모든 링크를 축약시킨 하위그래프와 싸이클에 포함되지 않는 링크들을 모두 축약시킨 하위그래프를 얻은 후, 각 하위그래프내에서 링크 이중장애에 따른 최대손실량을 구함으로써 G에 대한 최대손실량을 산출할 수 있다는 것이다. 이 싸이클 분해법은 이중링크 생존도의 대해서는 그 계산상의 복잡도가 $O(n^4)$으로 매우 효율적이며 시험계산결과를 통해서도 그 효율성이 입증되었다. 또한 단일링크장애에 따른 생존도의 계산에도 효율적으로 수정 적용될 수 있음이 밝혀졌다.

다만 이 싸이클 분해법은 그래프내에서 싸이클이라는 독특한 요소를 이용하는 방법이므로 오직 두 개 링크와 임의의 싸이클 사이의 관련성만을 이용하고 있다. 따라서 일반적인 개념하의 k-링크 장애를 고려하는 경우로 일반화될 수는 없다. 즉, 이중링크 컷과 임의의 싸이클과는 일정한 규칙을 가지지만, k-링크 컷과 싸이클과는 특정의 관련을 갖지 않는 것이다. 결국, k-링크 컷을 나열하기 위한 효과적인 방법은 보다 일반화된 그래프의 특성을 반영할 때 찾아질 수 있게 된다.

2.3 k다중연결 그래프에서의 최소 링크 컷의 도출

일반화된 k-링크 생존도는 그래프의 링크연결도와 관계된다. 주어진 그래프의 링크연결도가 k라고 하면, k-링크 생존도는 결국 최소 링크연결도를 규정하는 링크집합을 나열하

는 문제로 귀착되기 때문이다. 그래프의 링크연결도는 임의의 두 노드간에 존재하는 중복되지 않는 경로의 최소 개수로 정의되며, 이를 산출하기 위한 많은 연구들이 제시되어 있다(Ball & Provan, 1983, Nagamochi, Sun & Ibaraki, 1991, Nagamochi & Ibaraki, 1992)

특히, Nagamochi와 Ibaraki(1992)는 그래프의 연결도를 효과적으로 찾아내는 효율적인 알고리즘을 제시하였다 기존의 방법들이 노드간의 흐름량을 구하는 문제를 반복적으로 적용하여 그래프의 링크연결도를 산출하는 방법이었는 데에 반하여 이들이 제시한 방법은 걸침나무(spanning tree)를 구하는 단순한 알고리즘을 반복적으로 적용하고, 이로부터 얻어진 정보를 이용하여 링크를 축약시켜 가는 과정을 통하여 주어진 그래프의 링크연결도를 구할 수 있도록 고안되었다 알고리즘의 효율측면에서도 기존의 알고리즘보다 우수하며, 또 이 방법을 통하여 최소링크연결도를 규정하는 링크집합을 찾을 수 있음도 보이고 있다

한편 Ball과 Provan(1983)은 방향성이 존재하는 링크로 구성되는 그래프에서 최소링크 컷을 모두 나열하는 방법을 제시하고 있다 본 연구에서는 Nagamochi와 Ibaraki가 제시한 링크연결도 산출과정과 Ball과 Provan이 제시한 최소링크 컷을 나열하는 과정을 복합적으로 적용하여 k-링크 생존도 산출알고리즘을 제시한다 다만, 현실적으로 k-링크장애를 고려하는 경우, 해당그래프의 $(k-1)$-링크 생존도는 충분히 크다($S_{k-1}(G)=1$)고 전제할 수 있으므로 대상이 되는 그래프 G는 k-다중연결 그래프임을 전제로 한다.

III. k-링크 생존도 산출법

주어진 그래프의 최소링크 연결도를 계산하고, 이후 이 최소링크 연결도를 규정하는 링크집합을 모두 찾아낼 수 있으면 k-링크 생존도를 구할 수 있다. 이에 본 연구에서는 Nagamochi와 Ibaraki의 연구에서 제시된 절차를 수정하여 일반적인 통신망에서 k-링크 생존도를 산출하는 절차를 제시한다 제시되는 절차는 4장에서 시험적용을 통하여 분석된다

3.1 그래프의 링크연결도 계산

링크연결도(link-connectivity)는 그래프상에 존재하는 임의의 두 노드를 연결하며 서로 중복되는 링크를 포함하지 않는 경로(path)의 최대 수를 의미한다 즉, 링크연결도가 k인 그래프에서는 어떠한 k-1개의 링크에 장애가 발생한다고 하더라도 그래프내의 모든 노드 사이를 연결해 주는 경로가 존재하게 되는 것이다

임의의 그래프가 주어졌을 때, 이 그래프의 링크연결도를 찾기 위한 연구들이 지속적으로 이루어졌다. 그 중의 많은 연구들은 최대흐름문제를 반복적으로 풀어서 링크연결도를 계산한다. 그러나 Nagamochi와 Ibaraki(1992)는 보다 효과적인 방법으로 링크연결도를 찾는 알고리즘을 제시하고 있다 즉, 주어진 그래프 G상의 링크들을 토대로 몇 개의 중복되지 않는 포리스트(forest)를 구성하고, 이들간의 상호관련성을 토대로 그래프의 링크연결도를 탐색하는 방법을 제시하였다. 본 연구에서는 이들이 제시한 방법론 중에서 링크의 구조적 특성을 파악하는 부분을 채택하고 이를 이용하여 그래프의 k-링크생존도 산출절차를

제시하였다.

이들이 제시한 그래프이론, 관련요소 및 상호관련성은 다음과 같다. 먼저 그래프 $G=(N,E)$에 대해서, $H_i=(N,E_i)$가 $G-(E_1\cup\cdots\cup E_{i-1})$의 최대 포리스트가 되도록 링크집합 $E_i(\,i=1,\cdots,|E|)$를 정의한다[1] 그러면 서로 다른 i, j에 대해서 $E_i\cap E_j=\emptyset$ 이며 $E_1\cup\cdots\cup E_{|E|}=E$이고, 어떤 i에 대해서는 $\cup_{j,j\geq i}E_j=\emptyset$ 이다. 또 그래프의 연결성과 관련되어 아래와 같은 성질을 만족한다

Lemma 3.1 $G_i=(N,\cup_{j,j\leq i}E_j)$라 하면, 서로 다른 $x,y\in N$에 대해서 다음이 성립한다.

$$\lambda(x,y,;G_i)\geq\min\{\lambda(x,y;G),i\}.$$

증명. Nagamochi & Ibaraki(1992)의 Lemma 3.1 참조

Lemma 3.1에 의해서 그래프 G가 k-다중연결 그래프이면, $\cup_{j,j>k}E_j$에 포함되는 링크를 모두 제외해도 G의 k-다중연결성은 보장된다. 따라서 $\cup_{j,j>k}E_j$에 속하는 링크들을 축약하는 것은 k-링크 생존도 계산을 단순화시키는 좋은 방법이 된다 즉, $E_1\cdots,E_k$에 속하지 않는 링크는 k-링크 생존도 계산에 영향을 미치지 않는다는 사실에 착안하여 $\{E_i|i\geq k+1\}$에 속하는 링크를 모두 축약함으로써 검토해야 하는 k-링크집합의 수를 줄일 수 있게 된다 이 과정을 '단순화 과정(Simplification)'으로 정의하여 k-링크 생존도 산출과정에 활용한다.

Theorem 3.2 $\cup_{j,j>k}E_j$에 속하는 링크들을 축약하여도 G의 최소링크 연결도는 k이다

증명· $\cup_{j,j>k}E_j$에 속하는 링크들을 모두 축약한 그래프는 $G_k=(N,\cup_{j,j\leq k}E_j)$가 되며, Lemma 3.1에 의하여 G_k의 링크연결도는 k이상이 된다.

또 이들이 제안한 링크 연결도 산출과정 Procedure TESTEC는 다음과 같이 정형화된다

```
Procedure TESTEC
begin
    G' := G;  k := +∞;
    while |N'| ≥ 3 in G' = (N', E') do
        begin
            Compute partition E_1, E_2, ..., E_|E'| of E' by applying FOREST
            Choose a node w ∈ N' with |E'(w)| = δ(G');
            If δ(G') < k then let F := E'(w);
            k := min{k, δ(G')},
```

1) Nagamochi와 Ibaraki는 이를 위한 FOREST라는 프로시저를 제시하고 있다

Let $G' := G' / \{u, v\}$ with a link $(u, v) \in E_{\delta(G')}$,

end

{ $|N'| = 2$ now holds, and hence $\lambda(G') = \delta(G') = |E'|$ }

If $\delta(G') < k$ then let $F := E'$,

$k = \min\{k, \delta(G')\}$;

Conclude that $\lambda(G) = k$ and F is a minimum cut of G

end

TESTEC에 의한 링크연결도 계산과정의 개요는 다음과 같다 먼저 그래프 G에 대해서 $\{E_i | i = 1, \cdots, |E|\}$를 구성한다 다음으로 degree가 최소인 임의의 노드 t를 선택하여, 노드 t의 degree를 감안하여 링크연결도를 조정한다. 마지막으로 t에 연결된 링크이면서, $E_{\delta(G)}$에 포함되는 링크를 축약시킨다 이 과정을 반복함으로써 그래프내의 최소링크 컷을 모두 탐색할 수 있게 되는 것이다

TESTEC는 근본적으로 최소링크 컷이 될 가능성이 있는 모든 컷을 탐색하게 된다 따라서 본 연구의 대상인 k-다중연결 그래프에 이 방법을 적용하게 되면, 모든 k-링크 컷을 탐색할 수 있게 되는 것이다 그러나 본 연구에서는 오직 k-링크 컷만을 모두 탐색하면 되므로, 앞에서 언급한 '단순화 과정'을 포함시키면 탐색하는 컷의 수를 많이 줄일 수 있다 한편, 링크 생존도를 계산하는 경우에는, 컷에 속하는 링크수만을 고려하는 링크연결도 계산 경우와는 달리, k-링크 컷의 장애에 따른 서비스의 단절량을 산출하는 하위모듈이 필요하게 된다

3.2 k-링크 생존도 산출 절차

링크연결도가 k인 그래프 G의 k-링크 생존도 산출과정은 '단순화과정'으로 시작된다 즉, Nagamochi등의 FOREST 알고리즘을 이용하여 $\{E_1, \cdots, E_{|E|}\}$를 구하고 $\bigcup_{i>k} E_i$에 속하는 링크들을 모두 축약시키는 것이다 $E_{k+1} = \emptyset$일 때까지 이 단순화과정을 반복한다

다음으로 E_k에 속하는 임의의 링크 l를 선택하고, 이 링크를 포함하는 모든 k-링크 컷을 대상으로 해당 컷의 동시장애에 의한 서비스 손실량을 계산하게 된다. 우선 링크 l의 양쪽 종단 노드를 각각 x와 y라 하자. 그러면 E_1에서 E_k까지의 포리스트를 이용하여 x에서 y까지를 연결하는 k개의 경로를 얻을 수 있다. 이후 이 k개의 경로에 포함되지 않는 링크들을 이용하여 x와 y를 연결하는 경로가 있는지를 확인한다. $k+1$번째의 경로가 존재할 때, 링크 l은 바로 축약되며, 그렇지 않을 때는 링크 l을 포함하는 k-링크 컷을 탐색한 후, 축약되게 된다

링크 l을 포함하는 k-링크 컷의 탐색은 프로시저 ANTICHAIN[2)]을 이용하였다. 먼저

x에서 y로의 k개의 경로에 포함되는 링크들은 그 흐름의 방향대로만 포함시키고, 그렇지 않은 링크들은 모두 축약시킨다. 이렇게 얻어진 그래프는 x를 출발점, y를 도착점으로 하며, 싸이클을 포함하지 않는 방향성 그래프가 된다. 이제 프로시져 ANTICHAIN을 이용하여 링크 l을 포함하는 모든 컷을 찾을 수 있고, 이들 컷의 동시적 장애로 인한 서비스의 단절량을 계산할 수 있다 참고로 찾아진 컷에 따른 서비스 단절량의 계산은 최대 $|T|$번의 덧셈으로 이루어진다.

선택된 링크 l에 대한 탐색이 끝난 후 링크 l을 축약하게 되며, 결국 그래프 G의 노드 수는 하나 줄어들게 된다. 따라서 최대 $|N|-1$번의 링크탐색/축약과정을 반복함으로써, 모든 k-링크 컷의 장애에 따르는 서비스 단절량을 산출할 수 있으며 결과적으로 k-링크 생존도를 구할 수 있게 된다.

지금까지 제안된 알고리즘은 아래의 프로시져 k-SUR로 정형화 될 수 있다.

```
Procedure k-SUR
While |N|≥2 do
    begin
        Apply the 'Simplification' procedure;
        Compute partition E_1, E_2, ..., E_|E'| of E' by applying FOREST of G';
        Select a link l∈E_k and let x and y be its two end nodes;
        If there exist k+1 link-disjoint paths between x and y then contract l,
        otherwise apply ANTICHAIN to investigate k-link cuts including l, and
            contract link l;
    end
```

한편 프로시져 k-SUR이 진행되는 동안에 이루어지는 링크 축약과 링크연결도 사이에는 다음과 같은 관계가 성립한다.

Theorem 3.3 그래프 G가 k-다중연결 그래프이면, G에 속하는 임의의 링크 l를 약시킨 G/l도 k-다중연결 그래프이다.

증명. 링크 l이 노드 x와 y를 연결하고 있다고 하자 그러면 그래프 G내의 또다른 노드 z와 x사이에는 k개의 경로가 있다. 이 중에서 하나의 경로가 링크 l을 포함하는 경우에, 해당 경로는 나머지 $k-1$개의 경로와 중복되지 않으며 z에서 y까지를 연결하고 있다. 따라서 링크 l이 축약되는 경우에도 z와 축약된 노드 $x+y$ 사이에는 k개의 경로가 유지되게 된다.

2) 프로시져 ANTICHAIN은 싸이클이 포함되지 않는 방향성 그래프에서 출발점과 도착점사이의 모든 컷을 찾아 주는 알고리즘으로 Ball & Provan에 의해서 제시되었다.

따라서 프로시저 k-SUR은 $|M|-1$번의 링크축약과 ANTICHAIN을 통한 $W_k(G, T)$계산과정을 통하여 k-링크 생존도 $S_k(G, T)$를 구하게 된다

3.3 알고리즘 k-SUR의 수리적 복잡도

k-SUR의 수리적 복잡도는 FOREST, 단순화과정, ANTICHAIN, 링크 축약 등에 의해서 규정되게 된다 프로시저 FOREST의 복잡도는 $O(|N|+|E|)$이며(Nagamochi & Ibaraki, 1992), 하나의 링크축약은 그래프의 구조조정과 그에 따른 서비스 요구의 종단노드 조정으로 이루어진다 따라서 그 복잡도는 $O(|E|+|T|)$로 규정된다. 한편 두 노드사이의 흐름량을 확인하는 과정은 모든 링크의 흐름용량이 1인 경우로 한정되기 때문에 $O(|N||E|)$로 규정될 수 있고, procedure ANTICHAIN을 통한 두 노드 x, y사이의 링크컷의 나열은, 그 링크컷의 수를 $C(x, y, G)$라 할 때, $O(|C(x, y, G)||E|)$의 시간 내에 구해질 수 있다(Ball & Provan, 1983) 또 확인된 컷의 장애에 따른 서비스 손실량의 계산은 $O(|T|)$시간 내에 이루어질 수 있다 결국 그래프 G에서 임의의 두 노드 사이의 컷의 최대 수를 $c(G)$라 하면, k-SUR의 계산상의 복잡도는 단순화과정을 포함시키지 않을 때, $O(c(G)|N||E||T|))$로 규정될 수 있다. 또 한번의 단순화과정은 한번의 FOREST 수행과 하나이상의 링크 축약으로 이루어지므로 단순화의 계산상의 복잡도는 $O(|N|+2|E|+|T|)$로 규정될 수 있으며, 이를 통하여 하나이상의 노드수가 줄어들게 되므로 단순화과정의 도입은 k-SUR의 계산상의 효율을 높여주게 된다.

IV. 시험적용 및 결과 분석

본 연구에서는 3장에서 제안된 알고리즘을 C언어로 코딩하여 펜티엄 PC에서 시험 적용하였다. 노드수 50에서 100개 사이의 그래프를 랜덤하게 만들고, 제시된 알고리즘을 이용하여 k-링크 생존도를 계산하였다. 또한 기존의 싸이클 분해법과의 효율성 비교를 위하여 이중링크 생존도 계산도 시험하였다

먼저 대상문제는 특별히 고안된 문제작성 프로그램을 통하여 만들어졌다 제안된 알고리즘이 현실에서도 충분히 활용될 수 있는지를 검증할 수 있도록 노드 수는 50, 80, 100개로 변화시켰다. k-다중연결 그래프를 만들기 위하여, 먼저 평면좌표상에 정해진 수만큼의 노드를 위치시킨 후, 걸침나무(spanning tree)가 되도록 링크를 연결시켰다. k가 2인 경우에는 Monma와 Schallcross(1989)에 의해서 제시된 "Two-tree Dense"라는 휴리스틱을 이용하여 링크를 추가하여 이중연결그래프를 만들었다. 또 k가 3이상인 경우에는 여기에 추가하여 노드간 연결도를 확인하면서 그 연결도가 k이상이 되도록 링크를 추가해갔다. k-다중연결 그래프가 완성된 이후, 연결된 링크의 수가 미리 정해진 링크수에 미치지 못할 때에는 아직 연결되지 않은 임의의 노드쌍을 추가링크로 연결하였다. 그래프의 모양이 만들어진 후, 정해진 만큼의 서비스 요구량을 임의의 노드간에 부과하였다.

또 상황적인 변화에의 적절한 대처능력을 확인하기 위하여 링크수와 서비스 요구의 수를 다양하게 하였다. 링크의 수는 그래프의 연결성을 보장할 수 있도록 노드수의 1.6배에서 3.2배까지로 변화를 시켰으며, 서비스 요구의 수는 노드수와 같은 수준에서 노드수의 3배까지로 다양하게 변화시켰다 알고리즘의 시험적용 결과는 같은 규모의 대상망 10개에 대한 평균의 개념으로 정리하였다.

이러한 기준에 근거한 시험적용결과를 두 가지 방향으로 정리하였다 먼저 다중연결 그래프의 생존도 계산이 제시된 알고리즘을 통하여 원활히 이루어질 수 있음을 보이기 위하여 k가 3이상일 경우를 대상으로 540문제를 풀어서 정리하였다. 정리된 결과표에서 |N|은 노드수, |E|는 링크수, |T|는 서비스 요구의 수를 각각 나타낸다. 또 S_k는 링크생존도, N_s는 단순화 횟수, N_c는 컷의 수, CPU는 알고리즘의 실행시간을 나타내며, <표 3>의 CPUc는 싸이클 분해법의 실행시간을 나타낸다.

<표 1>과 <표 2>에서 보는바와 같이 다양한 형태의 그래프에 대해서 본 연구의 알고리즘은 매우 효과적으로 적용될 수 있다 노드수 100개인 경우는 현실에서도 매우 큰 통신망으로 분류되지만, 이 경우에도 수초 내에 그 링크생존도를 구할 수 있는 것이다 다만, 시험적용에서 서비스 요구의 수는 현실문제에 비해서 상대적으로 적게 감안된 상태이며, 알고리즘의 실행시간은 서비스 요구의 수의 증가에 따라서 비례적으로 증가하는 점을 감안하면 보다 많은 서비스 요구가 주어지는 경우라도 무리 없이 적용될 수 있음을 알 수 있다.

다음으로 제시된 알고리즘의 효율성을 보여주기 위하여 k가 2일 때를 대상으로 270 문제를 풀어서 정리하였다. 이 경우는 본 연구의 알고리즘과 싸이클 분해법(Kim, 1995)과의 비교를 통하여 정리하였다. <표 3>에서 보는 바와 같이 이중링크 장애에 따른 링크 생존도의 산출에 있어서도 본 연구의 방법론이 효율적으로 사용될 수 있음을 알 수 있다. 다만 링크의 수가 상대적으로 적어서, 컷의 수가 많은 경우에는 싸이클 분해법이 보다 효율적임을 알 수 있다 이는 싸이클 분해법이 이중링크 컷의 특수한 성질을 보다 잘 반영하고 있기 때문이라 할 수 있다 그러나 본 연구의 알고리즘이 가지는 k의 크기에 관련 없이 적용될 수 있다는 장점을 고려할 때, <표 3>의 결과는 만족할 만하다고 하겠다

V. 결 론

본 연구에서는 통신망의 장애대비능력을 분석하는 방법을 제시하였다. k개의 전송링크에 동시적인 장애가 발생하는 경우에도 여전히 처리될 수 있는 비율로서 통신망의 k-링크 생존도를 정의하고, 이의 체계적인 산출방법을 제시하였다. 또한 제시된 방법론의 계산상의 복잡도를 분석하였으며, 랜덤하게 만들어진 문제를 대상으로 시험적용을 하여 그 결과를 정리하였다. 대상문제들은 현실의 통신망을 대표할 수 있을 정도의 규모이며, 계산결과는 본 연구의 방법론이 현실 통신망의 분석에도 효과적으로 활용될 수 있음을 보여주고 있다.

다만, 연구의 전제로서 링크의 전송용량 무제한성을 가정하고 있는데, 이로 인하여 실제적인 적용에 제약이 있게 된다. 즉, 정보전송의 시간지연이 어느 정도 인정되는 패킷

(packet)방식 통신망의 생존도 분석에는 직접적인 적용이 가능하지만, 본격적인 멀티미디어 서비스의 도입에 따라 정보전송의 실시간성이 중요해지는 미래 통신망의 생존도 분석에는 적합하지 못할 수 있는 것이다. 따라서 향후 설비의 처리용량을 감안한 통신망의 성능지표 분석 및 이를 토대로 한 설계방법론의 체계화 등에 보다 많은 연구가 집중되어야 할 것이다.

<參考文獻>

[1] Ball, M O and J S Provan, "Calculating bounds on reachability and connectedness in stochastic networks," *Networks* 13 (1983), 253-278

[2] Brush, G G. and N A Malrow, "Assuring the dependability of telecommunications networks and services," *IEEE Network magazine* Jan (1990) 29-34.

[3] Cardwell, R and G. Brush, "Meeting the challenge of assuring dependable telecommunications services in the '90s," *IEEE communications Magazine* (1990) 40-45

[4] Cardwell, R., C L Monma, and T Wu, "Computer-aided design procedures for survivable fiber optic networks," *IEEE J. SAC* 7 (1989), 1188-1197

[5] Gavish, B., P Trudeau, M Dror, M Gendreau and L Mason, "Fiberoptic circuit network design under reliability constraints," *IEEE J SAC* 7 (1989) 1181-1187.

[6] Grötschel, M., C L Monma, and M Stoer, "Facets for polyhedra arising in the design of communication networks with low-connectivity constraints," *SIAM Journal on Optimization* 2 (1992), 474-504

[7] Grötschel, M., C. L Monma, and M. Stoer, "Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints," *Operations Research* 40 (1992), 309-330

[8] Kim, H-J., "Quantitative survivability in communication networks," Journal of Management (University of Ulsan) 2-2 (1995), 1-16.

[9] Ko, C. W and C. L Monma, "Heuristic methods for designing highly survivable communications networks," Technical report, Bellcore 1989

[10] Monma, C L, B S Munson and W R Pullyblank, "Minimum-weighted two-connected spanning networks," *Mathematical Programming* 46 (1990) 153-171

[11] Monma, C. L. and D F Shallcross, "Methods for designing communications networks with certain two-connected survivability constraints," *Operations Research* 37 (1989), 531-541.

[12] Myung, Y.-S., H.-J Kim, and D-W Tcha, "Design of communication networks with survivability constraints," Under Revision in *Management Science*, 1997

[13] Nagamochi, H. and T Ibaraki, "A Linear-time algorithm for finding a sparse k-connected spanning subgraph of a *k*-connected graph," *Algorithmica* 7 (1992), 583-596.

[14] Nagamochi, H., Z. Sun, and T. Ibaraki, "Counting the number of minimum cuts in undirected multigraphs," *IEEE Trans. Reliability* 40 (1991), 610-614.

[15] Satyanarayana, A and M K. Chang, "Network reliability and the factoring theorem," *Networks* 13 (1983) 107-120.

[16] Swamy, M. N. S and K. Thulasiraman, *Graphs, networks, and algorithms*, John Wiley & Sons, Inc (1981)

[17] Wu, T., *Fiber network survivability*, Artech House, Boston 1992.

[18] Wu, T., D. J Kolar and R. C. Cardwell, "Survivable network architectures for broad-band fiber optic network: model and performance comparison," *J Lightwave Tech.* 6 (1988) 1698-1709

<표 1> 제안된 알고리즘의 시험적용 결과 (k=3인 경우)

|N|	|E|	|T|	서비스량	S_k	N_S	N_C	CPU*
50	100	50	2,491.2	0.884	6.4	31.1	0.18
		100	5,082.1	0.897	6.8	34.5	0.31
		150	7,701.6	0.926	6.5	32.5	0.40
	120	50	2,513.3	0.892	9.0	15.3	0.09
		100	5,008.2	0.911	10.4	14.8	0.14
		150	7,476.8	0.931	10.5	15.7	0.22
	140	50	2,540.6	0.914	6.6	5.2	0.01
		100	5,018.6	0.936	7.3	5.1	0.03
		150	7,575.2	0.936	6.8	6.2	0.02
80	160	80	4,176.5	0.913	13.4	53.0	0.73
		160	8,180.8	0.932	11.3	52.8	1.21
		240	11,967.2	0.941	13.0	51.9	1.82
	180	80	4,017.4	0.925	13.5	30.6	0.42
		160	8,121.4	0.945	11.9	30.7	0.70
		240	12,241.9	0.947	12.1	32.2	1.05
	200	80	4,167.4	0.931	14.7	18.8	0.27
		160	8,143.1	0.948	14.8	19.1	0.42
		240	12,003.0	0.945	15.6	19.0	0.68
100	200	100	5,065.6	0.933	14.9	64.5	1.33
		200	10,097.6	0.944	14.9	66.3	2.55
		300	15,222.4	0.939	12.8	65.5	3.62
	220	100	5,023.8	0.933	16.9	43.3	0.90
		200	9,985.4	0.950	16.1	45.3	1.61
		300	15,117.2	0.943	15.4	43.4	2.57
	240	100	5,101.4	0.938	16.8	30.7	0.70
		200	9,998.0	0.944	17.3	29.4	1.00
		300	15,364.0	0.957	14.6	28.2	1.45

* 자료의 입출력 시간은 제외되었음.
모든 지표는 10개 문제에 대한 적용결과의 평균임

<표 2> 제안된 알고리즘의 시험적용 결과 (k=4인 경우)

|N|	|E|	|T|	서비스량	S_k	N_S	N_C	CPU^*
50	150	50	2,534.2	0.890	11.1	27.2	0.21
		100	5,187.2	0.912	10.8	28.2	0.27
		150	7,558.5	0 919	9 0	29.0	0 39
	170	50	2,584.2	0.905	10 1	11.2	0 10
		100	5,102 6	0.919	10.1	10.4	0 11
		150	7,516 3	0 933	9.9	11 0	0 17
	190	50	2,515 9	0.922	5.4	3.9	0.01
		100	5,092.4	0.931	5 7	5.4	0 03
		150	7,563.2	0 938	5.3	4 0	0.02
80	230	80	4,091.8	0.918	16 1	58.8	1.03
		160	8,172 6	0.939	17 5	58.5	1.58
		240	11,968.8	0 945	16 3	58.5	2.11
	240	80	3,889.2	0.918	16.1	45.8	0 80
		160	8,074.0	0.939	15.7	45 4	1.16
		240	12,039.8	0 941	15.3	44.7	1.72
	250	80	3,919 4	0.930	16.4	34.4	0 64
		160	8,134.3	0 942	15.6	34.7	1.04
		240	11,905.8	0.927	16 8	34.4	1.33
100	280	100	5,225.3	0.934	21.6	87 9	2 46
		200	10,033 2	0.938	21.3	86.6	3.92
		300	15,030.6	0.948	21 1	85.5	5.56
	300	100	5,083 5	0.928	20 4	57.3	1 55
		200	4,761.0	0 928	25.0	64.0	1.71
		300	15,315.3	0.928	21 7	58 4	3.37
	320	100	5,016.8	0.938	18.9	40 9	1 15
		200	10,138.3	0.941	20.6	37 5	1.60
		300	15,088.0	0.950	19.0	35.9	2.41

* 자료의 입출력 시간은 제외되었음.
모든 지표는 10개 문제에 대한 적용결과의 평균임.

<표 3> 싸이클 분해법과의 비교 (k=2인 경우)

$\|N\|$	$\|E\|$	$\|T\|$	서비스량	S_k	N_C	CPU^*	CPU_C^*
50	80	50	2,574.6	0.863	18.4	0.07	0.01
		100	5,179.0	0.895	18.2	0.12	0.01
		150	7,596.9	0.867	18.5	0.18	0.03
	100	50	2,496.1	0.903	7.6	0.01	0.02
		100	5,243.0	0.916	8.5	0.02	0.01
		150	7,575.4	0.932	6.9	0.03	0.03
	120	50	2,602.2	0.945	2.4	0.01	0.04
		100	5,049.6	0.946	3.1	0.01	0.02
		150	7,577.8	0.939	2.5	0.01	0.04
80	130	80	4,072.8	0.900	27.5	0.26	0.03
		160	8,097.9	0.906	29.0	0.39	0.04
		240	12,112.8	0.832	29.8	0.61	0.10
	160	80	4,049.4	0.932	10.1	0.01	0.04
		160	8,289.8	0.950	9.3	0.04	0.08
		240	11,883.9	0.935	10.4	0.06	0.12
	190	80	4,040.4	0.953	4.8	0.01	0.04
		160	8,075.4	0.955	4.8	0.03	0.08
		240	12,275.3	0.940	4.4	0.02	0.11
100	160	100	4,957.5	0.915	40.4	0.50	0.04
		200	10,181.3	0.930	37.1	0.75	0.09
		300	15,236.1	0.931	36.3	1.26	0.18
	200	100	5,294.8	0.940	14.1	0.03	0.06
		200	10,032.1	0.955	15.3	0.07	0.11
		300	15,621.4	0.955	14.3	0.10	0.19
	240	100	5,116.0	0.957	6.6	0.01	0.07
		200	10,136.0	0.961	6.3	0.01	0.14
		300	15,495.2	0.963	5.2	0.03	0.22

* 자료의 입출력 시간은 제외되었음.
모든 지표는 10개 문제에 대한 적용결과의 평균임.